Remarks

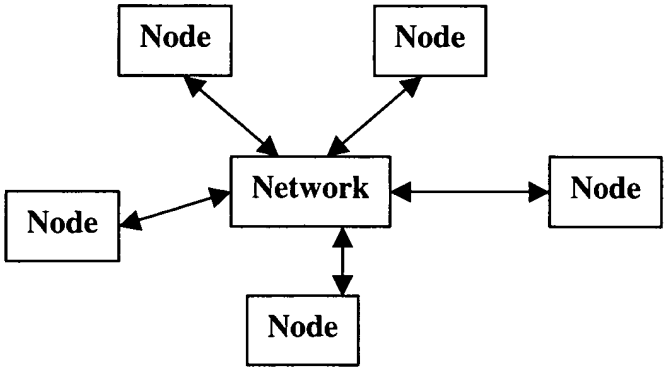
Claims 1-14 are pending in the application. Claims 1-14 are rejected.

Claims 1-3, 8, 9, 13 and 14 are rejected under 35 U.S.C. 102(b) as being anticipated by Erlichson et al (5,875,468).

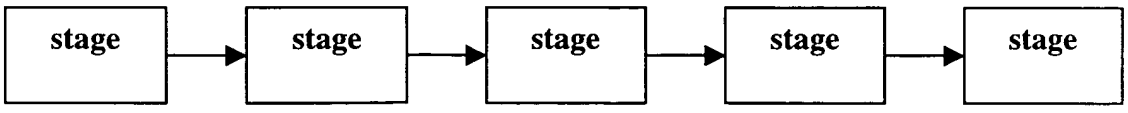
Erlichson describes pipelined *misses* when writing data to a cache. That is, **operations** (or methods) are pipelined.

The invention claims pipelined *stages*. That is, **structure** is pipelined.

Those of ordinary skill in the art would recognize the structure in Erlichson as a 'star' network, see Figure 4, or equivalently the figure below.



Claimed is a pipeline:



The network of nodes connected as a star in Erlichson can never anticipate pipelined stages connected serially.

The Erlichson reference was cited by the Applicants in the specification of the present application as an example of a pipeline ***method*** using a cache, see paragraph **[08]**:

"United States Patent 5,875,468, Erlichson, et al., February 23, 1999, "***Method*** to pipeline write misses in shared cache multiprocessor systems," describes a computer system with a number of nodes. Each node has a number of processors, which share a single cache. A ***method*** provides a release consistent memory coherency," emphasis supplied.

Applicants take note that a pipelined method as in Erlichson can never anticipate a pipelined structure as claimed. There is no relation.

In claim 1, a processing pipeline includes a plurality of stages connected serially to each other so that an output element of a previous stage is sent as an input element to a next stage, and a first stage is configured to receive input for a processing request, and a last stage is configured to produce output corresponding to the input.

Erlichson does not describe serially connected stages in this manner.

The Examiner's rejection addresses only three words of the first element of what is claimed, i.e. "*a processing pipeline.*" The Examiner is reminded that MPEP 2131 explicitly states that in order to anticipate a claim "each and every element as set forth in the claims" must be found in the prior art reference. "The identical invention must be shown in as complete detail as is contained in the ... claim." The Examiner points only to col. 5, lines 47-56 and Figure 4, below:

FIG. **4** shows a computer system **400** upon which the present invention may be practiced. Computer system **400** includes multiple processing nodes **401–406** for processing data according to a computer program. The present invention can be applied to systems having any number of nodes. Each of these nodes **401–406** includes one cache and one or more processors. These nodes **401–406** communicate with each other and associated shared elements via a general purpose interconnection network **407**.

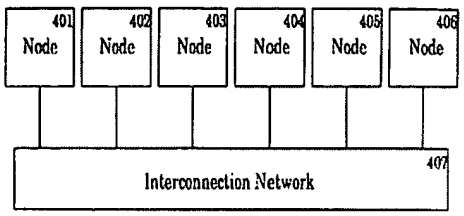


Figure 4

The Examiner is requested to specifically point out which words in the above section stages connected serially to each other so that an output element of a previous stage is sent as an input element to a next stage. There is no description of this explicit limitation, above. Further, a first stage is configured to receive input for a processing request, and a last stage is configured to produce output corresponding to the input. There are no identified first or last stages shown or described. Nothing in Figure 4 or its description above, anticipates what is claimed.

The structure of Erlichson is a 'star' not a pipeline. As known to those of ordinary skill in the art, with a star network, there is one common connection point for all nodes, i.e., the interconnect network in Figure 4. Figure 4 does not show or suggest a pipelined structure as claimed.

Further, claim 1 recites a progressive cache includes a plurality of caches ***arranged in an order*** from least finished cache elements to most finished cache elements, each cache for receiving an output cache element of a corresponding stage and for sending an input cache element to a next stage after the corresponding stage.

The Examiner again points to col. 5, lines 47-56, above, and asserts "*...each of the nodes 401-406 includes one cache.*" The Examiner's rejection fails

not only to provide a reasonable rational as to how, in the Examiner's view, the applied art can be construed to teach each and every feature in the rejected claims, but the rejection also fails to even consider explicitly claimed features of the invention as recited. The Examiner is requested to specifically point out which words mean a plurality of caches arranged in an order from least finished cache elements to most finished cache elements, each cache for receiving an output cache element of a corresponding stage and for sending an input cache element to a next stage after the corresponding stage. The Examiner's assertion is nothing more that an omnibus rejections and provides no reasonable level of understanding of the basis for the Examiner's position. As recognized in MPEP 707.07(d), "omnibus rejection of the claim ...is usually not informative and should therefore be avoided." Therefore, the rejection should be reconsidered and withdrawn.

There is no way that the nodes in Erlichson can be connected serially to form a processing pipeline.

Claim 1 also recites a cache controller is configured to route cache elements from the processing pipeline to the progressive cache in the order from a least finished cache element to a most finished cache element and from the progressive cache to the processing pipeline in the order from the most finished cache element to the next stage after the corresponding stage. The Examiner points to Erlichson at col. 6, lines 1-15, which describe Figure 5, both below:

write operations, processors **501–503** interact with cache memory **504**. Only if the data is not found in cache **504** (i.e., a cache miss), will a data transfer occur between cache memory **504** and local memory **507** or between cache memory **504** and remote caches or memory (not shown). A directory **505** is coupled to cache memory **504**. Directory **505** contains information regarding the status of the main memory, at cache line granularity: the directory keeps track of which caches in the larger system currently cache the data, and what the state is in each cache. Each cache-line-sized piece of main memory can be in one of three states, readable, writeable or invalid. An epoch and cache controller **506** is used to control the operation of the cache memory **504** and respond to the read, write, and synch operations of the processors **501–503**.

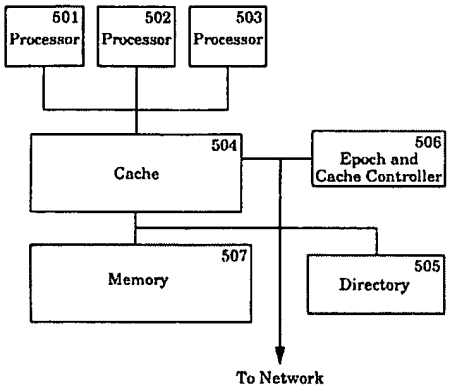


Figure 5

The Examiner again ignores the explicit limitations recited in claim 1. The Examiner asserts "*an epoch and cache controller 506 is used to control cache memory504.*" That teaches nothing recited in claim 1. There is nothing in the cited section about routing cache elements from the processing pipeline to the progressive cache. Nor is any order described, such as the claimed order from a least finished cache element to a most finished cache element and from the progressive cache to the processing pipeline in the order from the most finished cache element to the next stage after the corresponding stage. It appears that Erlichson fails to describe a single element of what is claimed in claims 1, 8, 13 or 14. Therefore, the Applicant's respectfully request the Examiner reconsider and withdraw his rejection based in Erlichson.

In claims 2 and 9, the progressive cache includes a cache for each stage of the processing pipeline. Erlichson only describes nodes having processors and a cache. As stated above, Erlichson never describes a plurality of stages connected serially to each other so that an output element of a previous stage is sent as an input element to a next stage, and a first stage is configured to

receive input for a processing request, and a last stage is configured to produce output corresponding to the input, as claimed. Therefore, Erlichson can never anticipate what is claimed.

In claim 3, the output cache element is stored in the corresponding cache. The claimed plurality of caches are arranged in an order from least finished cache elements to most finished cache elements. As stated above with respect to claim 1, there is no such order described by Erlichson. Further, in the invention each cache receives an output cache element of a corresponding stage and sends an input cache element to a next stage after the corresponding stage. The Examiner is requested to specifically point out where Erlichson teaches a plurality of stages connected serially to each other so that an output element of a previous stage is sent as an input element to a next stage. No such thing is described anywhere in Erlichson.

Claims 4-7, 10, 11 and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Erlichson et al as applied to claim 1, and further in view of Cheng et al (6,717,577).

Cheng describes a 3D graphics system where polygon vertex data is fed to a 3D graphics processor/display engine via a vertex cache used to cache and organize indexed primitive vertex data streams. The vertex cache fetches blocks of indexed vertex attribute data on an as-needed basis to make it available to the display processor.

Cheng fails to cure the defects of Erlichson. In particular, the vertex cache 212 cited by the Examiner is used only for input to the processing pipeline by the command processor 114, see Figure 1B, below:

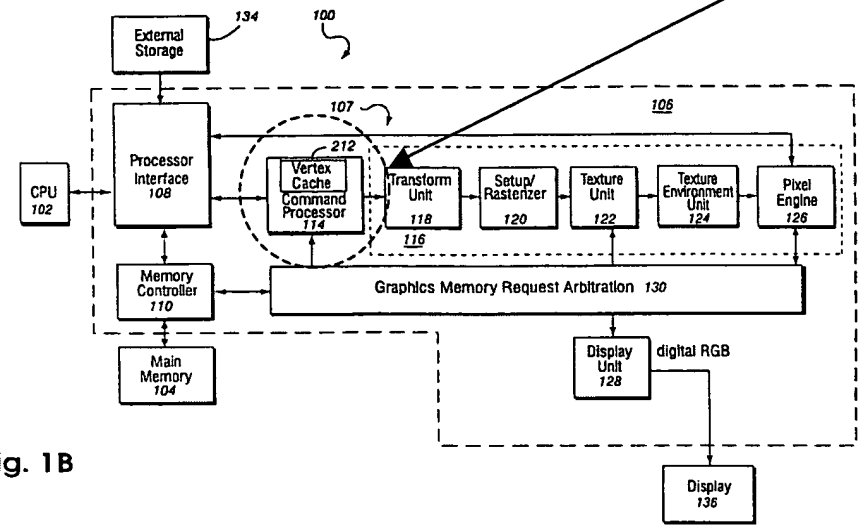


Fig. 1B

As stated above with respect to claim 1, the invention includes a progressive cache includes a plurality of caches ***arranged in an order*** from least finished cache elements to most finished cache elements, each cache for receiving an output cache element of a corresponding stage and for sending an input cache element to a next stage after the corresponding stage. A cache used only for input to a graphics pipeline as in Cheng can never make obvious what is claimed.

In claims 4 and 10, the cache elements are compressed. Each cache according to the invention receives an output cache element of a corresponding stage of the processing pipeline and sends an input cache element to a next stage after the corresponding stage. Therefore, cache elements in the pipeline can be compressed according to the invention. In contrast, Cheng's input-only vertex cache can receive compressed data, but

the data must be decompressed before it is fed to the processing pipeline, see col. 3, lines 58-66, below:

> In accordance with a further aspect provided by this invention, the vertex data includes quantized, compressed data streams in any of several different formats (e.g., 8-bit fixed point, 16-bit fixed point, or floating point). This data can be indexed (i.e., referenced by the vertex data stream) or direct (i.e., contained within the stream itself). These various data formats can all be stored in the common vertex cache, and subsequently decompressed and converted into a common format for the graphics display pipeline. Such hardware support of flexible types, formats and numbers of attributes

Cheng must decompress a data stream before inputting it to the processing pipeline. The invention compresses data between stages of the processing pipeline. Cheng can never make this obvious.

In claims 5 and 11, the cache elements are accessed by hashing. As stated above, Cheng only describes an input cache for a processing pipeline. A person of ordinary skill in the art would never confuse Cheng's input cache with the claimed progressive cache having associated processing stages. Neither of the references describes, alone or in combination, a processing pipeline and progressive cache having ordered caches associated with processing stages as claimed. Therefore, the Examiner's official notice that accessing caches using hashing is irrelevant. The same is true for claims 6 and 12, where least recently used cached elements are discarded when the progressive cache is full.

In claim 7, the input is a graphics object, and the output is an image. As stated above, the references, alone or in combination, fail to teach a processing pipeline, progressive cache, and cache controller as claimed.

Cheng processes graphics objects entirely differently than the invention and can never be used to make the invention obvious.

It is believed that this application is now in condition for allowance. A notice to this effect is respectfully requested. Should further questions arise concerning this application, the Examiner is invited to call Applicant's attorney at the number listed below. Please charge any shortage in fees due in connection with the filing of this paper to Deposit Account 50-0749.

Respectfully submitted,
Mitsubishi Electric Research Laboratories, Inc.

By _____

Andrew J. Curtin
Attorney for the Assignee
Reg. No. 48,485

201 Broadway, 8th Floor
Cambridge, MA 02139
Telephone: (617) 621-7573
Customer No. 022199